## if else

### **if**

if 语句语法格式：

if condition

then

command1

command2

...

commandN

fi

写成一行（适用于终端命令提示符）：

if [ $(ps -ef | grep -c "ssh") -gt 1 ]; then echo "true"; fi

末尾的fi就是if倒过来拼写，后面还会遇到类似的。

### **if else**

if else 语法格式：

if condition

then

command1

command2

...

commandN

else

Command

fi

### **if else-if else**

if else-if else 语法格式：

if condition1

then

command1

elif condition2

then

command2

else

commandN

fi

## for 循环

与其他编程语言类似，Shell支持for循环。

for循环一般格式为：

for var in item1 item2 ... itemN

do

command1

command2

...

commandN

done

写成一行：

for var in item1 item2 ... itemN; do command1; command2… done;

例如，顺序输出当前列表中的数字：

for loop in 1 2 3 4 5

do

echo "The value is: $loop"

done

顺序输出字符串中的字符：

for str in 'This is a string'

do

echo $str

done

输出结果：

This is a string

## while 语句

while循环用于不断执行一系列命令，也用于从输入文件中读取数据；命令通常为测试条件。其格式为：

while condition

do

Command

done

例：

#!/bin/bash

int=1

while(( $int<=5 ))

do

echo $int

let "int++"

done

运行脚本，输出：

12345

### **无限循环**

无限循环语法格式：

while :do

Command

done

或者

while true

do

Command

done

或者

for (( ; ; ))

## until 循环

until 循环执行一系列命令直至条件为 true 时停止。

until 语法格式:

until condition

do

Command

done

## case

Shell case语句为多选择语句。

case 值 in

模式1)

command1

command2

...

commandN

;;

模式2）

command1

command2

...

commandN

;;

esac

case工作方式如上所示。所有命令开始执行直至 ;;。

下面的脚本提示输入1到4，与每一种模式进行匹配：

echo '输入 1 到 4 之间的数字:'

echo '你输入的数字为:'

read aNum

case $aNum in

1) echo '你选择了 1'

;;

2) echo '你选择了 2'

;;

3) echo '你选择了 3'

;;

4) echo '你选择了 4'

;;

\*) echo '你没有输入 1 到 4 之间的数字'

;;

esac

## 跳出循环

Shell使用两个命令来实现该功能：break和continue。

### **break命令**

break命令允许跳出所有循环（终止执行后面的所有循环）。

#!/bin/bash

while :do

echo -n "输入 1 到 5 之间的数字:"

read aNum

case $aNum in

1|2|3|4|5) echo "你输入的数字为 $aNum!"

;;

\*) echo "你输入的数字不是 1 到 5 之间的! 游戏结束"

break

;;

esac

done

### **continue**

continue命令仅仅跳出当前循环。

对上面的例子进行修改：

#!/bin/bash

while :do

echo -n "输入 1 到 5 之间的数字: "

read aNum

case $aNum in

1|2|3|4|5) echo "你输入的数字为 $aNum!"

;;

\*) echo "你输入的数字不是 1 到 5 之间的!"

continue

echo "游戏结束"

;;

esac

done